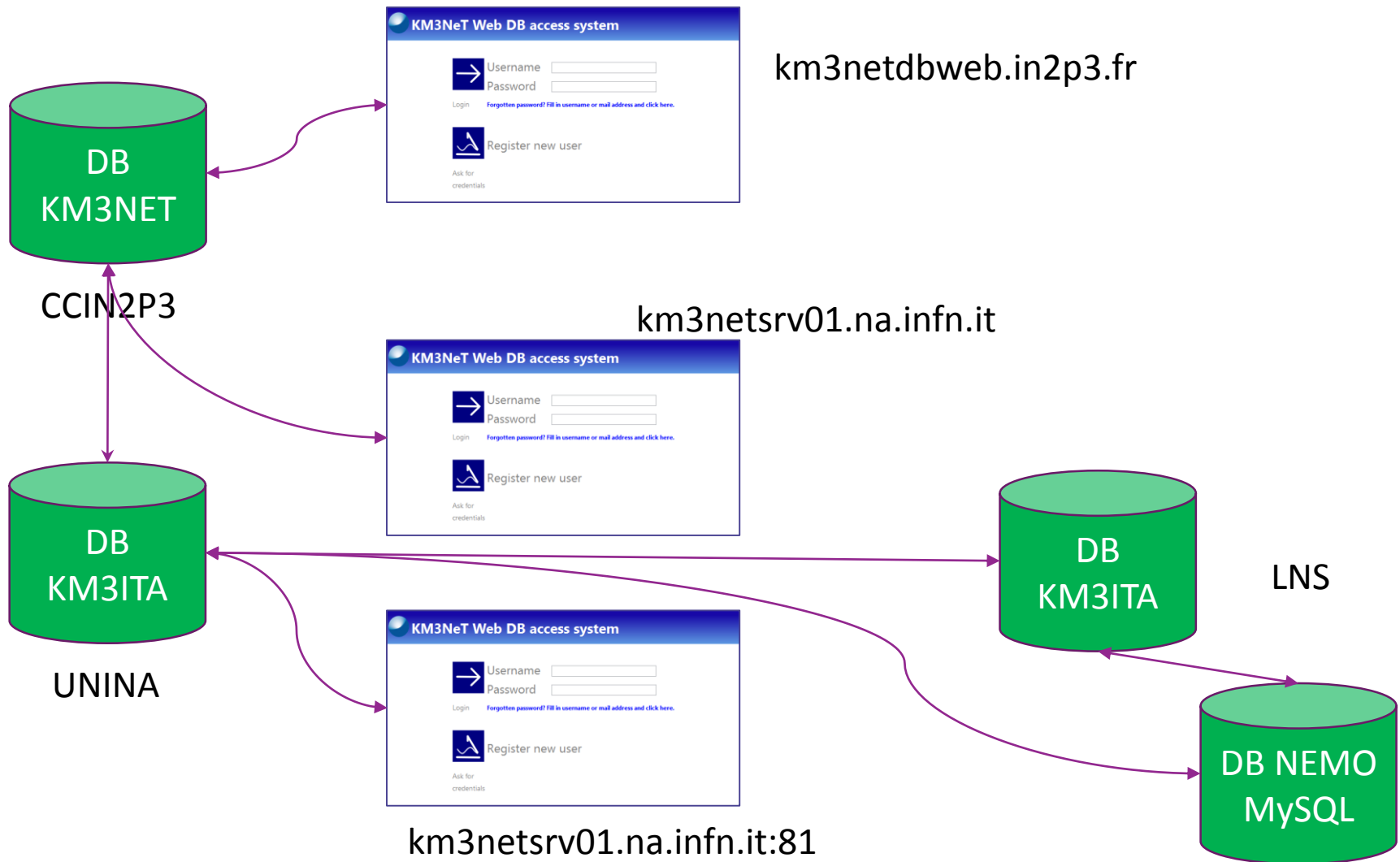


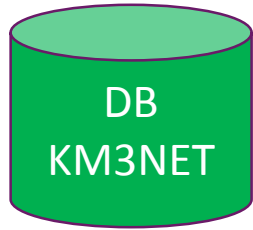
DATABASES
DB INTERFACE
GRAPHICAL INTERFACES

DATABASE SERVICES IN KM3NET

DB System in KM3NeT

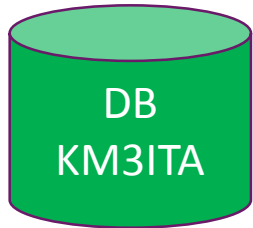


DB System in KM3NeT



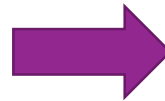
CCIN2P3

Oracle Database Server
Enterprise Edition
RAC (3 servers in SCAN)
700 GB (extendable)



UNINA

Oracle Database Server
Standard Edition
Expires July 2015



Sized to host several TB's/year
of Slow Control / DAQ Control data



Oracle Database Server
Enterprise Edition
Partitioning Option
100 NUPs Perpetual
~12 TB already available

Oracle licenses purchased with special conditions
 $100 \text{ NUPs} = 25 \text{ users} \times 8 \text{ cores} \times 0.5 \text{ (Intel core factor)}$

State of DB's

- CCIN2P3

- KM3NeT Collaboration management info (including authentication)
- PBS
- Integration/cabling information
- Products
- Test/calibration results
- PPM-DOM and PPM-DU book-keeping
- Detector definitions
- Runsetups
- Slow control / DAQ control data log
- In progress:
 - Paper authorship information
 - QA/QC documents

- KM3Ita

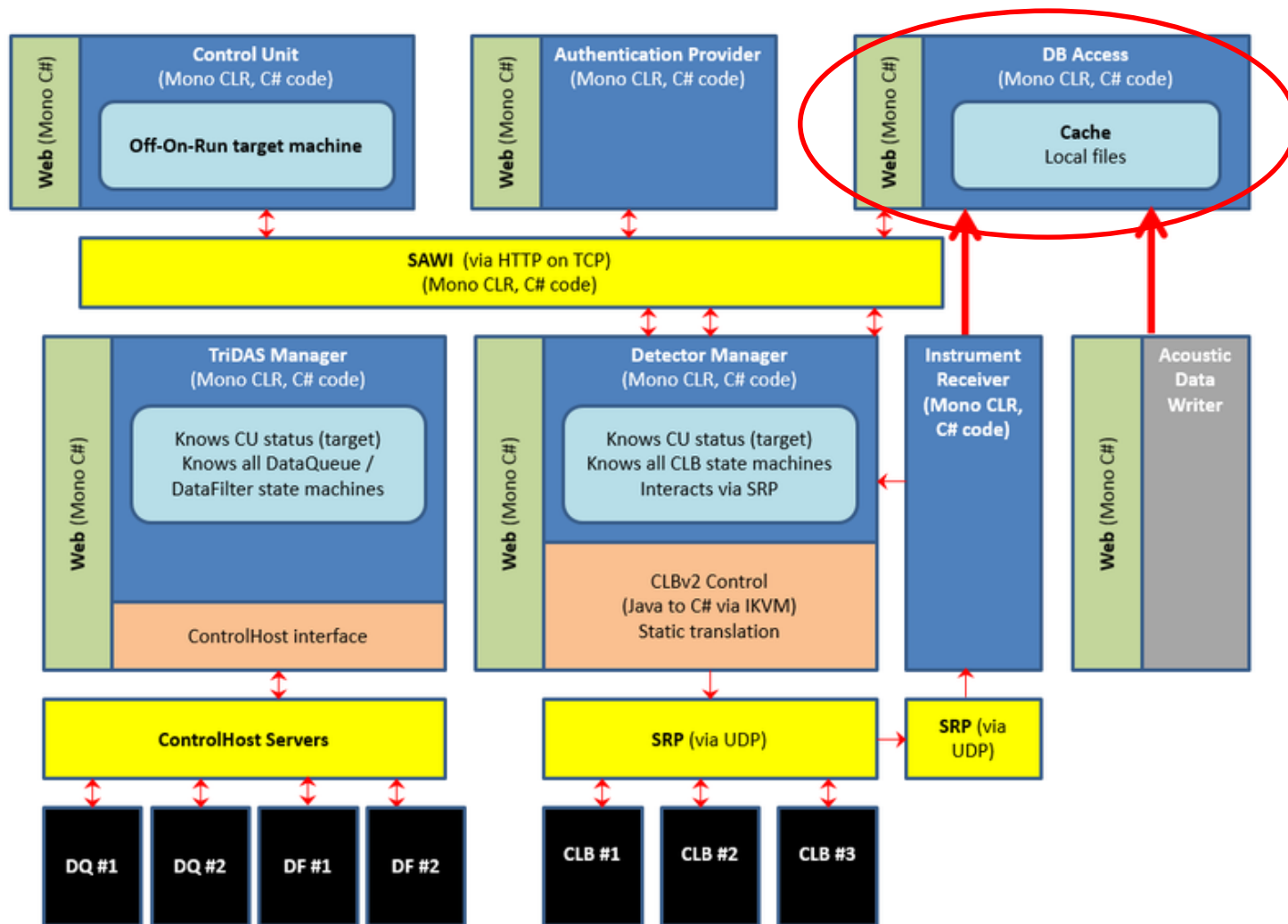
- User accounts sync'ed with CCIN2P3
- Products (in progress)
- Test/calibration results (in progress)
- Under development:
 - Runsetups
 - Slow control / DAQ control data log

- Complete dataset of NEMO phase 2 Slow Control
 - Copy from MySQL DB
- GUI interface to upload component documentation and test/calibration data in Excel format or ASCII format (PSS)
- A fraction of components is already documented, more to come soon

Functions of the DB Interface (by Arnould Albert)

- Provide access to the DB
 - HTTP protocol
 - Optimized SQL (read) and DML (write)
- Local read/write cache managed and optimized
 - Fault-tolerance with respect to network failures (works even if the central DB is temporarily unreachable)
 - Hides network latency
- Authentication services
 - Unified login credentials provided by central DB
 - Local encryption

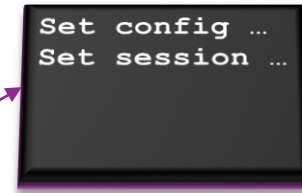
DB Interface – DB access to support acquisition



Detector Manager Structure

Web Server
SAWI, GUI (HTML, JSON)

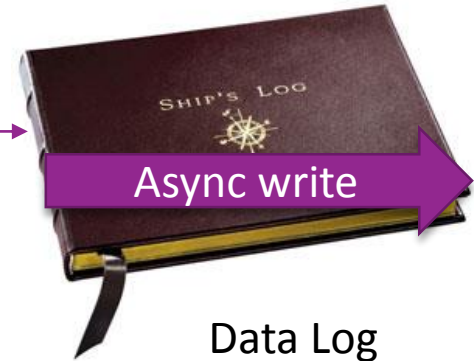
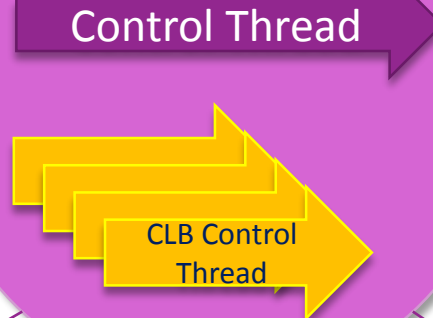
Control Core



Command console

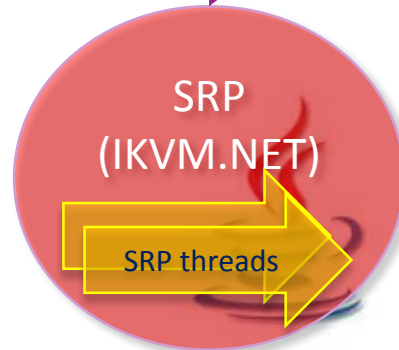
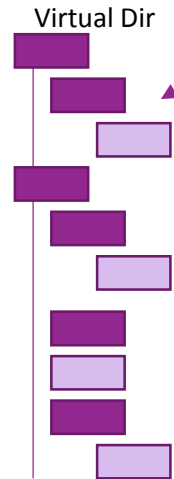


Notifications
from other
components
of the DU

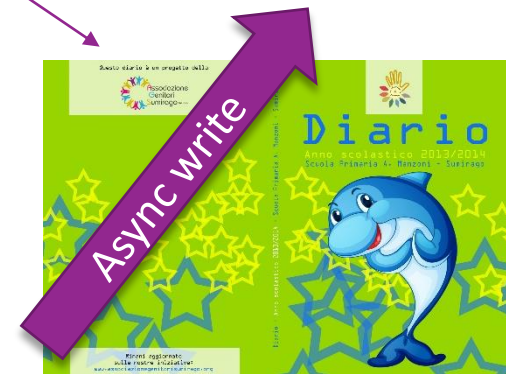


Async write

Data Log

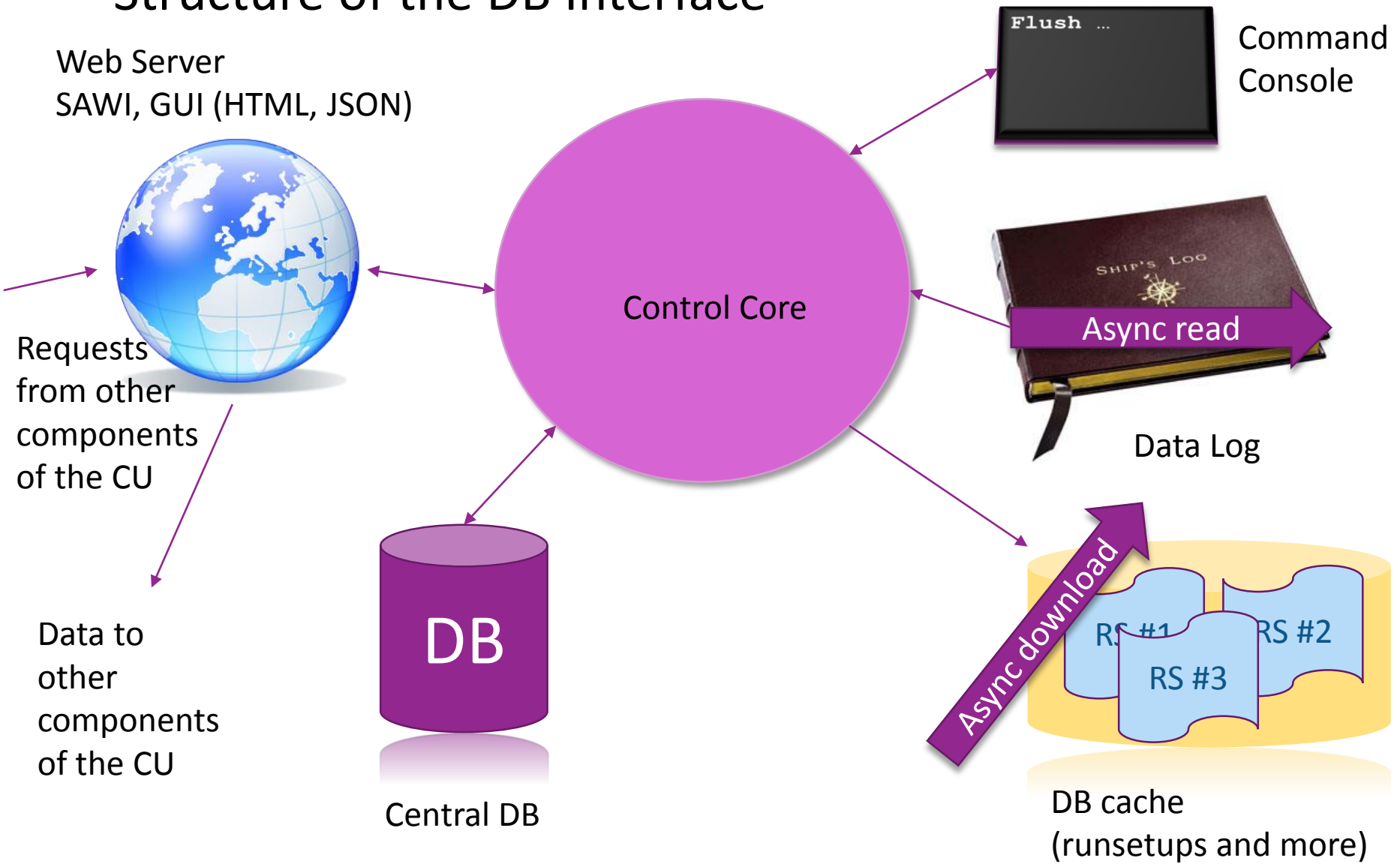


SRP to CLB

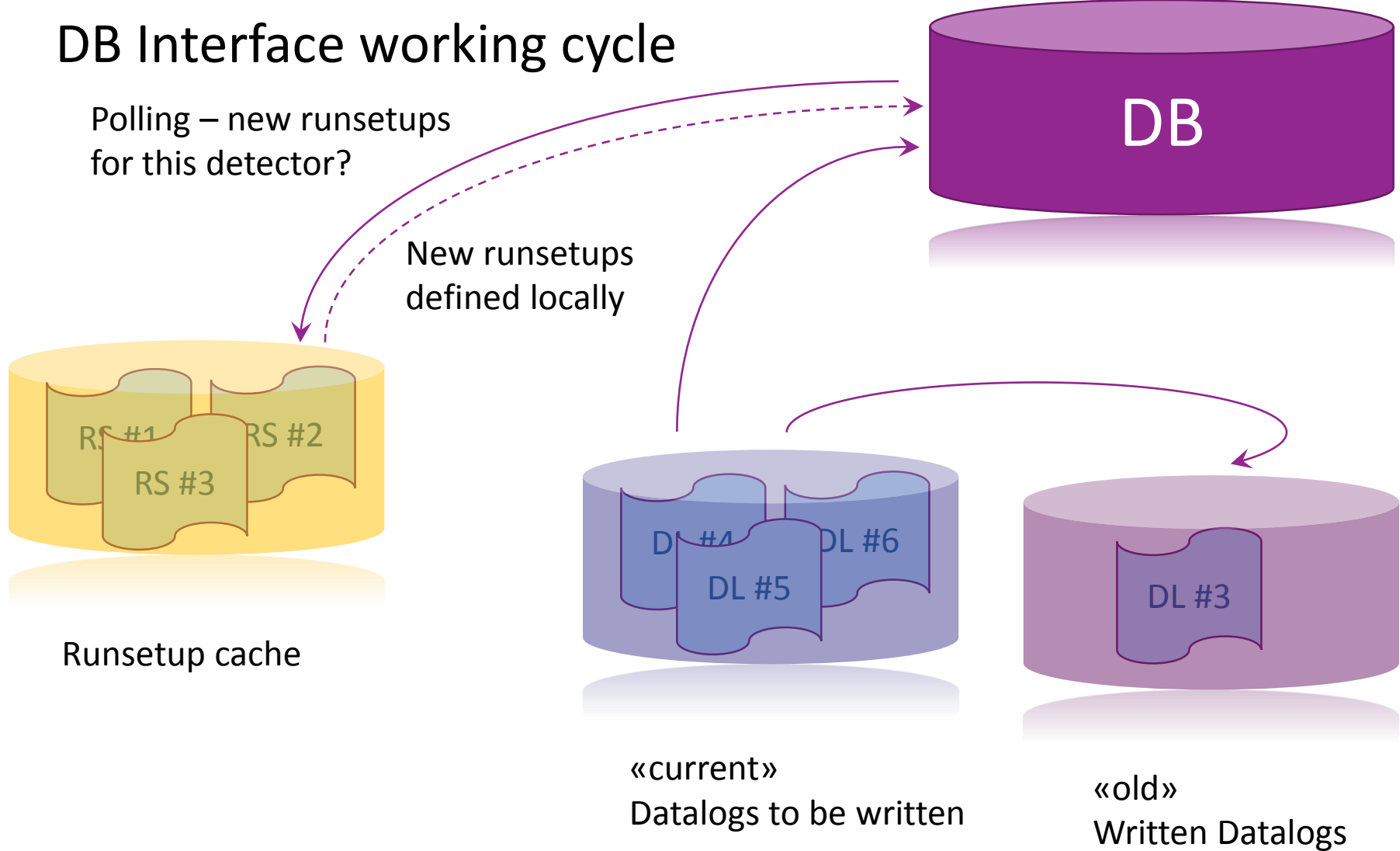


Text log

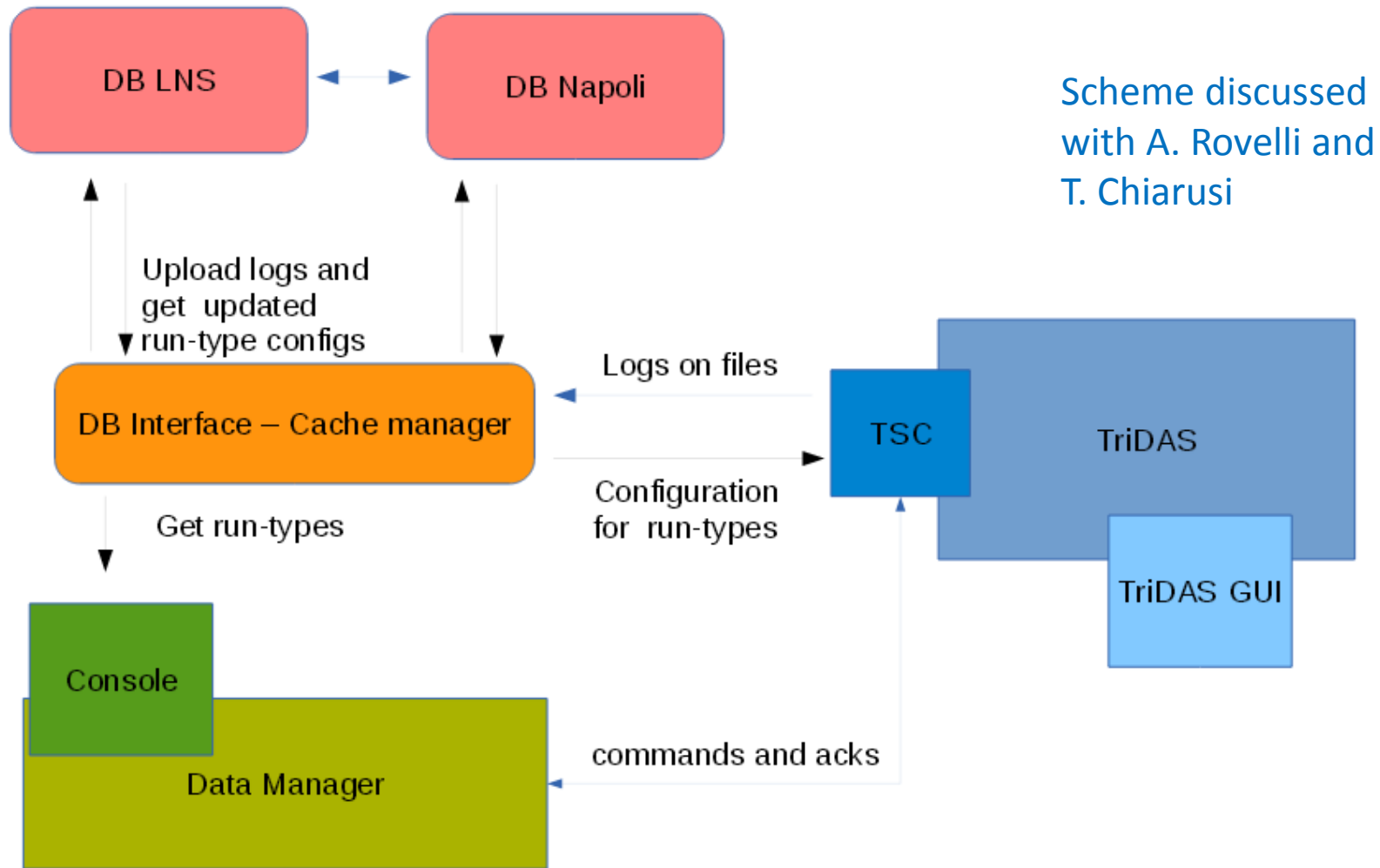
Structure of the DB Interface



DB Interface working cycle



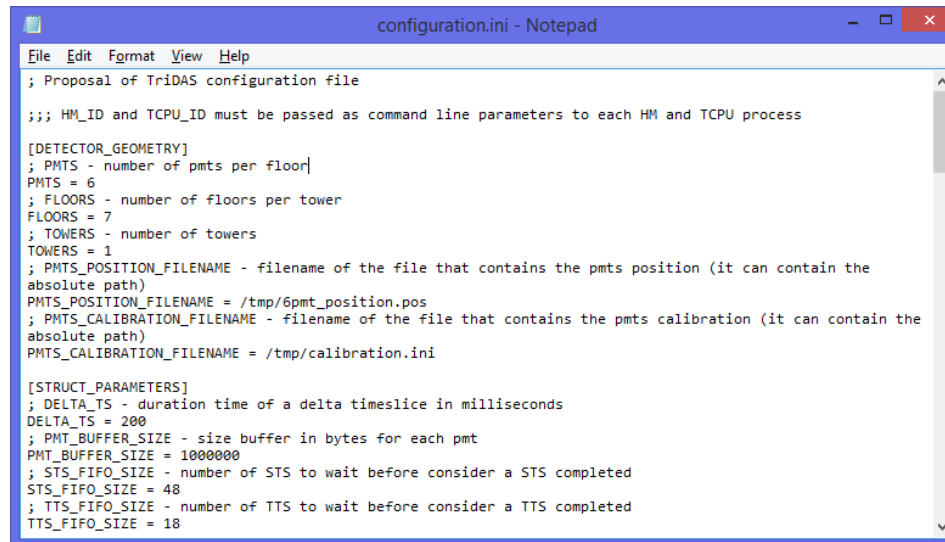
DB Interface in the towers: re-use SW from strings



11

DB Interface in the towers: re-use SW from strings

- Runsetup set for Data Manager and TriDAS
 - Format definition underway: Matteo Manzali



```
configuration.ini - Notepad
File Edit Format View Help
; Proposal of TriDAS configuration file

;;; HM_ID and TCPU_ID must be passed as command line parameters to each HM and TCPU process

[DETECTOR_GEOMETRY]
; PMTS - number of pmts per floor
PMTS = 6
; FLOORS - number of floors per tower
FLOORS = 7
; TOWERS - number of towers
TOWERS = 1
; PMTS_POSITION_FILENAME - filename of the file that contains the pmts position (it can contain the
absolute path)
PMTS_POSITION_FILENAME = /tmp/6pmt_position.pos
; PMTS_CALIBRATION_FILENAME - filename of the file that contains the pmts calibration (it can contain the
absolute path)
PMTS_CALIBRATION_FILENAME = /tmp/calibration.ini

[STRUCT_PARAMETERS]
; DELTA_TS - duration time of a delta timeslice in milliseconds
DELTA_TS = 200
; PMT_BUFFER_SIZE - size buffer in bytes for each pmt
PMT_BUFFER_SIZE = 1000000
; STS_FIFO_SIZE - number of STS to wait before consider a STS completed
STS_FIFO_SIZE = 48
; TTS_FIFO_SIZE - number of TTS to wait before consider a TTS completed
TTS_FIFO_SIZE = 18
```

- Uploading TriDAS logs into DB (technology similar to CU Datalogs)

Requirements for graphical interfaces

- Matteo Favaro volunteered
- Technology: HTML+Javascript+AJAX+JSON
 - Ease of integration into web applications (central webserver, DetectorManager, DBInterface, TriDAS Manager)
- Building runsetup/detector descriptions by GUI
- Ease of use
- Completeness
- Importing-reusing runsetups from DB or local cache

Web server for graphical interfaces

- Same technology as central Web services and CU
- Access and management of runsetup will be identical in central Web services and in shore stations
 - Local and remote handling of runsetups
- JSON protocol
- Functions to implement
 - List
 - Read
 - Write